JOURNAL OF COMPUTER AND SYSTEM SCIENCES 27, 1–13 (1983)

Tests for the *LR*-, *LL*-, and *LC*-Regular Conditions

STEPHAN HEILBRUNNER

Hochschule der Bundeswehr München, Fachbereich Informatik, Werner–Heisenberg-Weg 39, D 8014 Neubiberg, Germany

Received June 30, 1981; revised December 22, 1981

The original test for the *LR*-regular property is not quite correct. Generalizing the item method which is well known from $LR(k)$ theory a decidable criterion and a parsing algorithm are obtained. The method can be applied to *LL*-regular and *LC*-regular parsing too. It yields tests and inclusion theorems for the various classes of grammars considered.

1. INTRODUCTION

Most of the linear time parsing strategies (e.g., $LL(k)$ and $LR(k)$ type parsers) for context-free grammars operate by looking ahead on the input tape for a fixed number of symbols. The fixed length look-ahead strings partition the set of input strings into classes of strings which are equivalent with respect to parsing decisions. A moment's thought shows that these look-ahead classes are regular sets. This observation lends itself to a generalization introduced by Culik and Cohen [2]. The idea is to allow arbitrary sets as look-ahead classes as long as they form a finite partition of the set of input strings. Culik and Cohen applied this idea to *LR* parsing and obtained the class of *LR*-regular grammars. Later, the idea was applied to strong *LL* parsing [7] and to *LL* parsing [8]. In all cases tests and parsing algorithms for these grammar classes together with various other properties were obtained.

This paper had its origin in an example grammar where the criterion of Culik and Cohen fails. This example raises two questions. What is the class of grammars characterized by Culik and Cohen's criterion? How can we test for the *LR*-regular condition? Both questions will be answered in the sequel. The first problem will be solved by a simple trick. The answer to the second question will be obtained by generalizing the item method which is well known from $LR(k)$ theory. We shall make use of ideas of [4] and shall obtain tests and various other properties for the classes of *LL*-regular and *LC*-regular grammars as a by-product.

We establish our notation. "Grammar" always means context-free grammar. All grammars are supposed to be reduced, i.e., all nonterminals are reachable and produce terminal strings. The grammar $G = (V, \Sigma, P, S)$ is arbitrary but fixed. We shall make liberal use of the definitions, notations, and facts from [3]. Our conventions for the use of variables are shown in Table I, where Π denotes partitions of Σ^*. These conventions are an essential part of propositions, e.g., $L(G) = \{x \mid S \overset{*}{\Rightarrow} x\}$

0022-0000/83 $3.00

TABLE I

Variable	$a, b, \ldots$	$\ldots, y, z$	$\alpha, \beta, \ldots$	A	$\ldots, Y, Z$	$A, B, \ldots$	π
Values	Σ	Σ^*	V^*	V^0	V	N	Π

means $L(G) = \{x \in \Sigma^* \mid S \overset{*}{\Rightarrow} x\}$. To ease the burden of notation we usually omit outmost universal quantifiers, e.g.,

$$x \equiv y > zx \equiv zy \qquad \text{means} \quad \forall x\, \forall y\, \forall z\colon (x \equiv y > zx \equiv zy). \tag{+}$$

Notions concerning partitions are summarized now; $\Pi(w)$ is the equivalence class (block) containing w; Π is a left congruence if (+) holds; Π is regular if all $\pi \in \Pi$ are regular; Π' is a refinement of Π if $x \equiv y \bmod \Pi'$ implies $x \equiv y \bmod \Pi$. Every finite regular partition has a finite regular left congruent refinement which can be found effectively.

We extend $\equiv$ to sets $L', L'' \subseteq \Sigma^*$ via

$$L' \equiv L'' \bmod \Pi \;\Leftrightarrow\; \exists x \in L'\, \exists y \in L''\colon x \equiv y \bmod \Pi.$$

We note

$$L' \equiv L'' \bmod \Pi \;\Leftrightarrow\; \exists \pi \in \Pi\colon L' \cap \pi \neq \varnothing \wedge L'' \cap \pi \neq \varnothing.$$

In general, $\equiv$ is not (!) an equivalence relation on sets.

2. The $LR(\Pi)$ Definition

Two definitions from [2] are needed for the example announced in the Introduction.

Definition 2.1. G is CC–$LR(\Pi)$ if $S \Rightarrow_R^+ S$ is impossible and if

$$S \underset{R}{\overset{*}{\Longrightarrow}} \alpha A w \underset{R}{\Longrightarrow} \alpha\beta w \wedge S \underset{R}{\overset{*}{\Longrightarrow}} \gamma B x \underset{R}{\Longrightarrow} \gamma\delta x = \alpha\beta y \wedge w \equiv y \bmod \Pi \tag{1}$$

implies $A \to \beta = B \to \delta \wedge \alpha = \gamma \wedge x = y$. Let $\mathscr{P}$ be a partition of V^*. G is $LRRC(\mathscr{P}, \Pi)$ if $S \Rightarrow_R^+ S$ is impossible and if

$$S \underset{R}{\overset{*}{\Longrightarrow}} \alpha A w \underset{R}{\Longrightarrow} \alpha\beta w \wedge S \underset{R}{\overset{*}{\Longrightarrow}} \gamma B x \underset{R}{\Longrightarrow} \gamma\delta x = \rho\beta y \tag{2}$$

and

$$\alpha\beta \equiv \rho\beta \bmod \mathscr{P} \wedge w \equiv y \bmod \Pi \wedge |\rho\beta| \leqslant |\gamma\delta| \tag{3}$$

imply $A \to B = B \to \delta \wedge \gamma = \alpha \wedge x = y$. ∎

The condition on lengths for *LRRC* grammars is familiar from the $BRC(m, k)$ conditions (see, e.g., [1]) and will be seen to have a fairly natural explanation. Our example will show that the following proposition (see [2, Theorem 4.1.1) is not quite correct.

$$G \text{ is } CC\text{–}LR(\Pi) \quad \text{iff} \quad G \text{ is } LRRC(\mathscr{P}, \Pi) \text{ for some finite, regular } \mathscr{P}. \tag{4}$$

EXAMPLE 2.2. G has the following rules: $S \to ab$, $S \to Abc$, $A \to a$. We define the partition Π by

$$\Pi = \{\{c, A\}, \{b\}, \{bc\}, \{a, b, c\}^* \backslash \{A, c, b, bc\}\}$$

and shall prove that G is not $LR(\Pi)$ but $LRRC(\mathscr{P}, \Pi)$ for all partitions $\mathscr{P}$ of V^*. G admits the following three rightmost derivations:

$$S \underset{R}{\overset{\circ}{\Longrightarrow}} S \underset{R}{\Longrightarrow} ab, \tag{5}$$

$$S \underset{R}{\overset{\circ}{\Longrightarrow}} S \underset{R}{\Longrightarrow} Abc, \tag{6}$$

$$S \underset{R}{\Longrightarrow} Abc \underset{R}{\Longrightarrow} abc. \tag{7}$$

Table II gives an exhaustive case analysis of pairs of different derivations which satisfy condition (2). Line (5) + (7) violates the $CC\text{–}LR(\Pi)$ condition but satisfies the $LRRC(\mathscr{P}, \Pi)$ condition independent of the choice of $\mathscr{P}$. ∎

The test for the $CC\text{–}LR(\Pi)$ condition given in [2] relies substantially on proposition (4) (see [2, Corollary 5.8]). The test does characterize $LRRC(\mathscr{P}, \Pi)$ grammars, i.e., [2, Corollaries 5.6, 5.7] are certainly correct. Consequently, it does not characterize $CC\text{–}LR(\Pi)$ grammars. The obvious question is how to repair

TABLE II

Derivations		α	β	w	γ	δ	x	ρ	y	$w \equiv y$?	$\lvert\rho\beta\rvert \leqslant \lvert\gamma\delta\rvert$?
(5)	(6)	A	ab	A	A	Abc	A	—	—	—	—
(5)	(7)	A	ab	A	A	a	bc	A	c	Yes	No
(6)	(5)	A	Abc	A	A	ab	A	—	—	—	—
(6)	(7)	A	Abc	A	A	a	bc	—	—	—	—
(7)	(5)	A	a	bc	A	ab	A	A	b	No	Yes
(7)	(6)	A	a	bc	A	Abc	A	—	—	—	—

proposition (4). It is easy to see that it is the length condition of the $LRRC$-definition which invalidates (4). Thus, we arrive at

DEFINITION 2.3. A grammar is $LR(\Pi)$ if $S \Rightarrow_R^+ S$ is impossible and if

$$S \underset{R}{\overset{*}{\Longrightarrow}} \alpha A w \underset{R}{\Longrightarrow} \alpha\beta w \wedge S \underset{R}{\overset{*}{\Longrightarrow}} \gamma B x \underset{R}{\Longrightarrow} \gamma\delta x = \alpha\beta y \tag{8}$$

and

$$w \equiv y \bmod \Pi \wedge |\alpha\beta| \leqslant |\gamma\delta| \tag{9}$$

imply $A \to \beta = B \to \delta \wedge \alpha = \gamma \wedge x = y$. ∎

THEOREM 2.4. *Let Π be a finite regular partition. G is $LR(\Pi)$ iff G is $LRRC(\mathscr{P}, \Pi)$ for some finite regular partition $\mathscr{P}$ of V^*.*

Proof. See [2, Theorem 4.1] and note that the length condition in the "<" direction causes no problems with our $LR(\Pi)$ definition. ∎

THEOREM 2.5. *Every CC–$LR(\Pi)$ grammar is $LR(\Pi)$. There is a partition Π and a grammar which is $LR(\Pi)$ but not CC–$LR(\Pi)$. If Π is a left congruence, then a grammar is $LR(\Pi)$ iff it is CC–$LR(\Pi)$.*

Proof. See Example 2.2 for the second claim. The third claim is easily verified (see [1, Lemma 5.2, or 3, Lemma 12.2.3] for the $LR(k)$ case). ∎

There is an alternative, less commonly used $LR(k)$ condition which uses certain regular sets (see, e.g., [1, Excercise 5.2.10]). We shall generalize this definition for later use and thereby obtain another justification of our $LR(\Pi)$ definition.

DEFINITION 2.6. For each $A \to \beta \in P$ define its reduction context by

$$RC(A \to \beta) = \{\alpha\beta \# w \mid S \underset{R}{\overset{*}{\Longrightarrow}} \alpha A w \underset{R}{\Longrightarrow} \alpha\beta w\},$$

where $\#$ is a new symbol, $\# \notin V^*$. In addition, let

$$RC(P) = \bigcup_{A \to \beta \in P} RC(A \to \beta). \quad ∎$$

The next theorem is a generalization of the alternative $LR(k)$ condition and is easily verified (see [4] for the $LR(k)$ case). Note that the $LR(k)$ case defines Π by

$$w \equiv x \bmod \Pi \Leftrightarrow k{:}\, w = k{:}\, y,$$

where $k{:}\, w = u$ if $(|u| = k \wedge \exists v{:}\, w = uv)$ and $k{:}\, w = w$ if $|w| < k$.

THEOREM 2.7. *G is LR(Π) iff it satisfies* (10) *and* (11) *and does not allow a derivation* $S \Rightarrow_R^+ S$.

$$\psi \# w \in RC(A \to \beta) \wedge \psi \# x \in RC(B \to \delta) \wedge w \equiv x \bmod \Pi > A \to \beta = B \to \delta, \tag{10}$$

$$\psi \# w \in RC(P) \wedge \psi t \# x \in RC(P) \wedge w \equiv tx \bmod \Pi > t = \Lambda. \quad \blacksquare \tag{11}$$

The proof relies on the length condition in (9). Since (10) and (11) arise naturally in connection with the usual shift–reduce parsing algorithm the length condition in (9) is not as strange as it may seem at first sight. The next theorem characterizes *CC–LR(Π)* grammars in terms of reduction contexts. It is easily verified if careful attention is paid to the position of the marker # and to the tricky equivalence condition in (11).

THEOREM 2.8. *G is CC–LR(Π) iff G is LR(Π) and satisfies* (12).

$$\rho r \# w \in RC(P) \wedge \rho \# ry \in RC(P) \wedge w \equiv y \bmod \Pi > r = \Lambda. \quad \blacksquare \tag{12}$$

If Π is a left congruence, $w \equiv y$ implies $rw \equiv ry$ and then (12) is a consequence of (11).

3. ITEM GRAMMARS

In order to test for the conditions (10)–(12) we introduce a generalized version of those items which are well known from *LR(k)* theory. Most of the proofs for generalized items are very similar to the corresponding proofs for items as presented in [4] so that proofs will be mostly omitted.

A (*generalized*) *item* is either the symbol $[S]$ or is a quadruple written as $[A \to \alpha \cdot \beta, \pi]$, where $A \to \alpha\beta \in P$ and $\pi \in \Pi$. We use $[\ ]$, $[\ ]'$,..., as variables for items. Recall that $\Pi(\Lambda)$ is the equivalence class which contains Λ.

DEFINITION 3.1. The *item grammar* for G and Π is defined by $G_I = (I \cup V, V, P_I, [S])$, where I is the set of items for G and Π and

$$\begin{aligned} P_I = {} & \{[S] \to [S \to \cdot\, \delta, \Pi(\Lambda)] \mid S \to \delta \in P\} \\ & \cup \{[A \to \alpha \cdot X\beta, \pi] \to X[A \to \alpha X \cdot \beta, \pi] \mid [A \to \alpha \cdot X\beta, \pi] \in I\} \\ & \cup \{[A \to \alpha \cdot B\beta, \pi] \to [B \to \cdot\, \delta, \pi'] \mid L(\beta)\,\pi \cap \pi' \neq \emptyset \\ & \quad \wedge [A \to \alpha \cdot B\beta, \pi] \in I \wedge [B \to \cdot\, \delta, \pi'] \in I\}. \quad \blacksquare \end{aligned}$$

The item grammar is right linear and can be computed effectively whenever Π is finite and regular because in this case $L(\beta)\,\pi \cap \pi'$ is a context-free language (see, e.g., [3, Theorem 6.4.1]) whose emptiness problem is solvable. We need a technical lemma relating a grammar and its reduction contexts to its item grammar.

LEMMA 3.2 (The Item Lemma). *Let Π be a left congruence.*

$$\exists j\colon [\ ] \overset{j}{\Rightarrow} \gamma[\ ]' \wedge |\gamma| = j \wedge [\ ] \neq [S] \not\equiv \exists A, \alpha, \beta, \pi\colon$$
$$[\ ] = [A \to \alpha \cdot \gamma\beta, \pi] \wedge [\ ]' = [A \to \alpha\gamma \cdot \beta, \pi], \tag{13}$$

$$\exists j\colon [A \to \alpha \cdot \beta, \pi_A] \Rightarrow^j \gamma[C \to \rho \cdot \sigma, \pi_C] \wedge |\gamma| \neq j$$
$$\not\equiv \exists r \in \pi_A\ \exists t \in \pi_C\ \exists \omega\colon \beta r \underset{R}{\overset{*}{\Longrightarrow}} \omega C t \underset{R}{\Longrightarrow} \omega\rho\sigma t = \gamma\sigma t, \tag{14}$$

$$[S] \overset{*}{\Rightarrow} \gamma[C \to \rho \cdot \sigma, \pi] \not\equiv \exists t \in \pi\ \exists \omega\colon S \underset{R}{\overset{*}{\Longrightarrow}} \omega C t \underset{R}{\Longrightarrow} \omega\rho\sigma t = \gamma\sigma t, \tag{15}$$

$$[S] \overset{*}{\Rightarrow} \psi[C \to \gamma \cdot, \pi] \not\equiv \exists t \in \pi\colon \psi \neq t \in RC(C \to \gamma). \quad \blacksquare \tag{16}$$

Proof. We illustrate the effect of equivalence classes by proving ">" of (15) and refer the reader to [4] for the proof of similar propositions, otherwise. The case $[S] \Rightarrow^1 \gamma[C \to \rho \cdot \sigma, \pi]$ entails $\gamma = A$, $[C \to \rho \cdot \sigma, \pi] = [S \to \cdot\, \sigma, \Pi(A)]$ and, hence, is trivial. In the case

$$[S] \Rightarrow [S \to \cdot\, \delta, \pi_S] \overset{*}{\Rightarrow} \gamma[C \to \rho \cdot \sigma, \pi_C]$$

either (13) or (14) is applicable. Condition (13) is trivial. Consider (14). We obtain

$$\delta r \underset{R}{\overset{*}{\Longrightarrow}} \omega C t \underset{R}{\Longrightarrow} \omega\rho\sigma t = \gamma\sigma t \qquad \text{for some} \quad r \in \pi_S, \quad t \in \pi_C.$$

Obviously, $t = sr$ and $\delta \Rightarrow_R^* \omega C s$ for some $s \in \Sigma^*$. Note $\pi_S = \Pi(A)$ so that $r \equiv A$ and $t = sr \equiv sA = s$, i.e., $t \equiv s$ and $s \in \pi_C$. Putting these together yields

$$S \underset{R}{\Longrightarrow} \delta \underset{R}{\overset{*}{\Longrightarrow}} \omega C s \underset{R}{\Longrightarrow} \omega\rho\sigma s = \gamma\sigma s \qquad \text{and} \qquad s \in \pi_C. \quad \blacksquare$$

The item lemma applies immediately to (10–(12) of Theorems 2.7 and 2.8 if Π is a left congruence. Otherwise, a little trick is needed. Let Π' be a refinement of Π, which can be found effectively for finite, regular Π. Each equivalence class of Π is the union of some equivalence classes of Π'. Moreover, for $\pi', \pi'' \in \Pi'$

$$\pi' \equiv \pi'' \bmod \Pi \not\equiv \exists \pi \in \Pi\colon \pi' \cup \pi'' \subseteq \pi$$

which reduces to

$$\pi' \equiv \pi'' \bmod \Pi \not\equiv \pi' = \pi''$$

if $\Pi = \Pi'$. In the sequel, Π' is to be a left congruent refinement of Π, and the item grammar is taken with respect to Π' (not Π).

LEMMA 3.3. *Propositions* (17)–(19) *are equivalent to* (10)–(12), *respectively.*

$$[S] \overset{*}{\Rightarrow} \psi[A \to \beta \cdot, \pi_A] \wedge [S] \overset{*}{\Rightarrow} \psi[B \to \delta \cdot, \pi_B] \wedge \pi_A \equiv \pi_B \bmod \Pi > A \to \beta = B \to \delta, \tag{17}$$

$$[S] \overset{*}{\Rightarrow} \psi[A \to \alpha \cdot a\beta, \pi_A] \wedge [S] \overset{*}{\Rightarrow} \psi[B \to \delta \cdot, \pi_B] > L(a\beta)\,\pi_A \not\equiv \pi_B \bmod \Pi, \tag{18}$$

$$[S] \overset{*}{\Rightarrow} \rho r[A \to \beta \cdot, \pi_A] \wedge \pi \equiv \pi_A \bmod \Pi \wedge r \neq A > RC(P) \cap \rho \neq r\pi = \varnothing. \tag{19}$$

Proof. We only show "(18) > (11)" and leave the other cases to the reader. Suppose

$$\psi \neq w \in RC(B \to \delta) \wedge \psi ar \neq y \in RC(C \to \gamma) \wedge w \equiv ary \bmod \Pi.$$

Then, by the item lemma

$$[S] \overset{*}{\Rightarrow} \psi[B \to \delta \cdot, \Pi'(w)] \wedge [S] \overset{*}{\Rightarrow} \psi ar[C \to \gamma \cdot, \Pi'(y)].$$

The structure of the item grammar yields an item such that

$$[S] \overset{*}{\Rightarrow} \psi[A \to \alpha \cdot a\beta, \pi_A] \overset{*}{\Rightarrow} \psi ar[C \to \gamma \cdot, \Pi'(y)].$$

The item lemma gives us $s \in \pi_A$, $y' \in \Pi'(y)$ such that $a\beta s \Rightarrow_R^* ary'$. We conclude successively:

$$y \equiv y' \bmod \Pi',$$

$$ary \equiv ary' \bmod \Pi',$$

$$ary \equiv ary' \bmod \Pi,$$

$$w \equiv ary' \bmod \Pi, \text{ because of } w \equiv ary \bmod \Pi,$$

$$L(a\beta)\,\pi_A \equiv \Pi'(w) \bmod \Pi \text{ because of } w \in \Pi'(w) \wedge ary' \in L(a\beta)\,\pi_A$$

which contradicts (18). ∎

Presently, we shall express conditions (17)–(19) with the help of certain sets. The intuition in defining these sets is derived from the usual shift–reduce algorithm. It is described by the following relation

$$(\alpha\beta, z) \vdash (\alpha A, z) \Join a\beta \neq z \in \text{reduce}(A \to \beta),$$

$$(\psi, az) \vdash (\psi a, z) \Join \psi \neq az \in \text{shift},$$

where

$$\text{reduce}(A \to \beta) = \{\psi \neq w \mid \exists \pi_A \in \Pi' \; \exists \pi \in \Pi: \\ [S] \overset{*}{\Rightarrow} \psi[A \to \beta \cdot, \pi_A] \wedge w \in \pi \wedge \pi_A \subseteq \pi\}$$

and

$$\text{shift} = \{\psi \neq w \mid \exists \pi \in \Pi \; \exists [A \to \alpha \cdot a\beta, \pi_A]: \\ [S] \overset{*}{\Rightarrow} \psi[A \to \alpha \cdot a\beta, \pi_A] \wedge L(a\beta)\, \pi_A \cap \pi \neq \varnothing \wedge w \in \pi\}$$

are the appropriate choices for $LR(\Pi)$ grammars. By straining intuition a little we obtain

$$CC\text{–shift} = \{\rho \neq rw \mid r \neq \Lambda \wedge \exists [A \to \beta \cdot, \pi_A] \; \exists \pi \in \Pi': (\pi \equiv \pi_A \bmod \Pi \\ \wedge w \in \pi \wedge [S] \overset{*}{\Rightarrow} \rho r[A \to \beta \cdot, \pi_A])\}$$

as the additional shift set for CC–$LR(\Pi)$ grammars.

LEMMA 3.4. *Propositions* (17)–(19) *are equivalent to* (20)–(22), *respectively:*

$$\text{reduce}(A \to \beta) \cap \text{reduce}(B \to \delta) \neq \varnothing > A \to \beta = B \to \delta, \tag{20}$$

$$\text{reduce}(B \to \delta) \cap \text{shift} = \varnothing, \tag{21}$$

$$PC(P) \cap CC\text{–shift} = \varnothing. \tag{22}$$

Proof. The proof is immediate. ∎

LEMMA 3.5. *Let Π' be finite, regular and left congruent. The sets* reduce$(A \to \beta)$, *shift, and* CC–shift *are regular and can be found effectively.*

Proof. We have

$$\text{reduce}(A \to \beta) = \bigcup_{\pi_A \in \Pi' \wedge \pi_A \subseteq \pi \in \Pi} \{\psi \neq \mid [S] \overset{*}{\Rightarrow} \psi[A \to \beta \cdot, \pi_A]\}\, \pi.$$

Let

$$L_1([A \to \alpha \cdot a\beta, \pi_A]) = \{\psi \neq \mid [S] \overset{*}{\Rightarrow} \psi[A \to \alpha \cdot a\beta, \pi_A]\},$$

$$L_2([A \to \alpha \cdot a\beta, \pi_A]) = \bigcup_{\pi_A \in \Pi \wedge L(a\beta)\pi_A \cap \pi \neq \varnothing} \pi_A.$$

Note that $L(a\beta)\,\pi$ is context-free so that $L(a\beta)\,\pi \cap \pi' \neq \varnothing$ is a decidable property. Now,

$$\text{shift} = \bigcup_{[A \to \alpha \cdot a\beta, \pi_A] \in I} L_1([A \to \alpha \cdot a\beta, \pi_A])\, L_2([A \to \alpha \cdot a\beta, \pi_A]).$$

Let

$$L_3([C \to \varphi \cdot a\psi, \pi_C]) = \{\rho \# \mid [S] \overset{*}{\Rightarrow} \rho[C \to \varphi \cdot a\psi, \pi_C]\}$$
$$L_4([C \to \varphi \cdot a\psi, \pi_C], [A \to \beta \cdot , \pi_A]) = \{r \mid [C \to \varphi \cdot a\psi, \pi_C] \overset{*}{\Rightarrow} r[A \to \beta \cdot , \pi_A]\}.$$

Then

$$CC\text{–shift} = \bigcup L_3([C \to \varphi \cdot a\psi, \pi_C])\, L_4([C \to \varphi \cdot a\psi, \pi_C], [A \to \beta \cdot , \pi_A])\, \pi,$$

where the union is taken over all $[C \to \varphi \cdot a\psi, \pi_C] \in I$, $[A \to \beta \cdot , \pi_A] \in I$, $\pi \in \Pi'$ such that $\pi \equiv \pi_A \bmod \Pi$. ∎

LEMMA 3.6. *$RC(P)$ is context-free and can be found effectively.* ∎

Proof. We adapt the proof of the same fact in [2, Lemma 5.1] to our terminology. Let $\{\mathbf{A} \mid A \in N\}$ be a new set of nonterminals corresponding to N in the obvious way. Define $\mathbf{a} = a$ for all $a \in \Sigma$ and $\boldsymbol{\alpha} = \mathbf{X}_1 \cdots \mathbf{X}_n$ if $\alpha = X_1 \cdots X_n \in V^*$. We modify the rules of the item grammar so that right context is retained and define

$$\begin{aligned}\mathbf{R} = {} & \{[S] \to [S \to \cdot\, \delta] \mid S \to \delta \in P\} \\ & \cup \{[A \to \alpha \cdot X\beta \to X[A \to \alpha X \cdot \beta] \mid A \to \alpha X\beta \in P\} \\ & \cup \{[A \to \alpha \cdot B\beta] \to [B \to \cdot\, \delta]\, \boldsymbol{\beta} \mid A \to \alpha B\beta, B \to \delta \in P\} \\ & \cup \{[B \to \delta \cdot] \to \# \mid B \to \delta \in P\} \\ & \cup \{\mathbf{A} \to \boldsymbol{\alpha} \mid A \to \alpha \in P\}.\end{aligned}$$

We claim $|S| \Rightarrow_{\mathbf{R}}^* \psi \# z \overset{\mathbf{X}}{\Rightarrow} \psi \# z \in RC(P)$. ∎

Combining these lemmas gives us

THEOREM 3.7. *For any grammar and any finite, regular partition Π the following questions are decidable: Is the grammar $LR(\Pi)$? Is the grammar $CC\text{–}LR(\Pi)$?* ∎

EXAMPLE 3.8. We return to the grammar of Example 2.2. The obvious choice for a left congruent refinement is $\pi' = \{\{\Lambda\}, \{b\}, \{c\}, \{bc\}, R\}$, where $R = \{a, b, c\}^* \setminus \{\Lambda, b, c, bc\}$. A short computation yields the following sets:

$D \to \delta$	$S \to ab$	$S \to Abc$	$A \to a$
reduce$(D \to \delta)$	$\{ab\#\}$	$\{Abc\#\}$	$\{a \# bc\}$

$$\text{shift} = \#R \cup \{a \# b, A \# bc, Ab \# c\},$$

$$CC\text{–shift} = \{\#ab, \#abc, a\#b, a \# bc, A \# bc, A \# bcc,$$
$$Ab \# c, Ab \# cc, \#abc\},$$
$$RC(P) = \{ab\#, Abc\#, a \# bc\}.$$

Now, $a \# bc \in RC(P) \cap CC$–shift. Hence, G is not CC–$LR(\Pi)$. ∎

Equipped with the generalized item grammar we can proceed along the lines of [4] and define the $LR(\Pi)$ automaton as the canonical deterministic automaton belonging to the item grammar for the left congruent refinement Π' of Π. This automaton is the "canonical collection of sets of items" together with the "GOTO-function" in the terminology of [1, 3]. Consistency of the $LR(\Pi)$ automaton is defined in the usual way, keeping in mind (17) and (18). Thus, the $LR(\Pi)$ automaton is consistent iff the grammar is $LR(\Pi)$. Moreover, the $LR(\Pi)$ automaton can be used in a generalized LR parsing algorithm which operates in linear time. The details of the theory are a straightforward excercise which has been worked out in [5].

4. $LL(\Pi)$ AND $LC(\Pi)$ GRAMMARS

In this section we report some results which can be obtained for $LL(\Pi)$ and $LC(\Pi)$ grammars with the help of the $LR(\Pi)$ automaton which was informally introduced in the last section. The proofs are straightforward (see [5]) generalizations of the proofs of analogous theorems of [4] and will be omitted.

Concerning LL-regular grammars, the obvious definition is [8]

DEFINITION 4.1. G is an $LL(\Pi)$ grammar if it satisfies the following condition:

$$S \underset{L}{\overset{*}{\Longrightarrow}} wC\omega \wedge C \to \gamma \in P \wedge C \to \delta \in P \wedge$$
$$L(\gamma\omega) \equiv L(\delta\omega) \bmod \Pi > \gamma = \delta. \quad ∎$$

The next theorem gives rise to a decision procedure if Π is a finite, regular left congruence.

THEOREM 4.2. *Let Π be a left congruence. G is $LL(\Pi)$ iff for each state q of its $LR(\Pi)$ automaton the following condition is satisfied:*

$$[C \to \cdot\, \gamma, \pi] \in q \wedge [C \to \cdot\, \delta, \pi'] \in q \wedge L(\gamma)\,\pi \equiv L(\delta)\,\pi' \bmod \Pi > \gamma = \delta. \quad ∎$$

A corollary to this theorem gives us a characterization of $LL(\Pi)$ grammars in terms of rightmost derivations.

COROLLARY 4.3. *Let Π be a left congruence. G is $LL(\Pi)$ iff*

$$S \underset{R}{\overset{*}{\Longrightarrow}} \psi A x \underset{R}{\Longrightarrow} \psi\beta x \wedge S \underset{R}{\overset{*}{\Longrightarrow}} \psi A y \underset{R}{\Longrightarrow} \psi\gamma y \wedge$$

$$L(\beta)\, x \equiv L(\gamma)\, y \bmod \Pi > \gamma = \delta. \quad \blacksquare$$

Unfortunately, the trick using left congruent refinements, working so well for $LR(\Pi)$ grammars, fails for $LL(\Pi)$ grammars so that a special test is needed. One may either turn to [8] or use the "local follow sets" known from $LL(k)$ theory. The following relation $\Rrightarrow$ formalizes this idea. Let Π' be a left congruent refinement of Π. For all $A, B \in N$ and $\Delta_1, \Delta_2 \subseteq \Pi'$ define

$$[A, \Delta_1] \Rrightarrow [B, \Delta_2] \Leftrightarrow \exists \gamma, \delta\colon A \to \gamma B \delta \in P \wedge$$

$$\Delta_2 = \{\pi \in \Pi' \mid \exists \pi' \in \Delta_1\colon L(\delta)\, \pi' \cap \pi \neq \varnothing\}.$$

THEOREM 4.4. *For all $C \to \gamma$, $C \to \delta \in P$*

$$\exists w\, \exists \omega\colon S \underset{L}{\overset{*}{\Longrightarrow}} wC\omega \wedge L(\delta\omega) \equiv L(\gamma\omega) \bmod \Pi$$

$$\Leftrightarrow \exists \Delta\, \exists \pi_1, \pi_2 \in \Delta\colon [S, \Pi'(\Delta)] \overset{*}{\Rrightarrow} [C, \Delta] \wedge L(\delta)\, \pi_1 \equiv L(\gamma)\, \pi_2 \bmod \Pi.$$

Obviously, $\Rrightarrow$ is a decidable relation so that the $LL(\Pi)$ condition is decidable for each $C \in N$, too.

Proof. The proof is based on

$$[S, \Pi'(\Delta)] \overset{*}{\Rrightarrow} [C, \Delta] \Leftrightarrow \exists w\, \exists \omega\colon S \underset{L}{\overset{*}{\Longrightarrow}} wC\omega \wedge$$

$$\Delta = \{\pi \in \Pi' \mid L(\omega) \cap \pi \neq \varnothing\}$$

which is shown by induction. $\blacksquare$

Is every $LL(\Pi)$ grammar $LR(\Pi)$?

EXAMPLE 4.5. Consider the grammar with the rules

$$S \to dAa, \qquad S \to dBb, \qquad A \to \Lambda, \qquad B \to \Lambda,$$

and define Π by

$$\Pi = \{\{da\}, \{db\}, \{a, b\}, \{a, b, d\}^* \backslash \{a, b, da, db\}\}.$$

The grammar is $LL(\Pi)$ because of $\{da\} = L(dAa) \not\equiv L(dBb) = \{db\}$ but not $LR(\Pi)$ because of $S \Rightarrow_R^* dAa \Rightarrow_R da \wedge S \Rightarrow_R^* dBb \Rightarrow_R db \wedge a \equiv b$. $\blacksquare$

THEOREM 4.6. *Let Π be a left congruence. If G is $LL(\Pi)$, then G is $LR(\Pi)$.* ∎

Turning to $LC(\Pi)$ grammars we use the following definition, which is the straightforward generalization of the $LC(k)$ definition used in [9].

DEFINITION 4.7. A grammar is $LC(\Pi)$ if it satisfies the following two conditions:

$$S \underset{R}{\overset{*}{\Longrightarrow}} \psi A w \underset{R}{\Longrightarrow} \psi X \beta w \wedge S \underset{R}{\overset{*}{\Longrightarrow}} \omega B y \underset{R}{\Longrightarrow} \omega \rho X \sigma y = \psi X \sigma y$$

$$\wedge\, L(\beta w) \equiv L(\sigma y) \bmod \Pi > \psi A = \omega B \wedge X\beta = \rho X \sigma,$$

$$S \underset{R}{\overset{*}{\Longrightarrow}} \psi A w \underset{R}{\Longrightarrow} \psi w \wedge S \underset{R}{\overset{*}{\Longrightarrow}} \omega B x \underset{R}{\Longrightarrow} \psi y \wedge w \equiv y \bmod \Pi$$

$$> x = y \wedge \psi A = \omega B. \quad ∎$$

For this definition we obtain the expected theorems.

THEOREM 4.8. *Let Π be a left congruence. G is $LC(\Pi)$ iff for each state q of the $LR(\Pi)$ automaton the following two propositions are true:*

$$[A \to \cdot\, X\beta, \pi] \in q \wedge [B \to \gamma \cdot X\delta, \pi'] \in q \wedge L(\beta)\,\pi \equiv L(\delta)\,\pi'$$

$$> [A \to \cdot\, X\beta] = [B \to \gamma \cdot X\delta]$$

$$[A \to \cdot\, \Lambda, \pi] \in q \wedge [D \to \cdot\, \delta, \pi'] \in q \wedge \delta \in \Sigma V^* \cup \{\Lambda\} \wedge \pi \equiv L(\delta)\,\pi'$$

$$> [A \to \cdot\, \Lambda] = [D \to \cdot\, \delta]. \quad ∎$$

THEOREM 4.9. *Let Π be a left congruence. Every $LL(\Pi)$ grammar is $LC(\Pi)$. Every $LC(\Pi)$ grammar is $LR(\Pi)$.* ∎

THEOREM 4.10. *Every $LC(\Pi)$ grammar has an equivalent $LL(\Pi')$ grammar if Π' is a common left congruent refinement of Π and $\{\{\Lambda\}\} \cup \{a\Sigma^* \mid a \in \Sigma\}$.* ∎

The proof of the last theorem is long and tedious. The main idea is to obtain the $LL(\Pi')$ grammar from an appropriate parsing algorithm for the $LC(\Pi)$ grammar and to use a different but equivalent $LC(\Pi)$ definition. Details are given in [5].

REFERENCES

1. A. V. AHO AND J. D. ULLMAN, "The Theory of Parsing, Translation and Compiling," Vol. I, Prentice–Hall, Englewood Cliffs, N. J., 1972.
2. K. CULIK II AND R. COHEN, *LR*-Regular grammars—an extension of $LR(k)$ grammars, *J. Comput. System Sci.* 7 (1973), 66–96.
3. M. A. HARRISON, "Introduction to Formal Language Theory," Addison–Wesley, Reading, Mass., 1978.

4. S. HEILBRUNNER, A parsing automata approach to *LR* theory, *Theoret. Comput. Sci.* **15** (1981), 117–157.
5. S. HEILBRUNNER, Zerteilungsverfahren mit unbeschränkter Vorschau, Bericht Nr. 8005, Hochschule der Bundeswehr München/Neubiberg, Germany, 1980.
6. J. E. HOPCROFT AND J. D. ULLMAN, "Introduction to Automata Theory, Languanges, and Computation," Addison–Wesley, Reading, Mass., 1979.
7. S. JARZABEK AND T. KRAWCZYK, *LL*-regular grammars, *Inform. Process. Lett.* **4** (1975), 31–37.
8. D. POPLAWSKI, On *LL*-regular grammars, *J. Comput. System Sci.* **18** (1979), 218–227.
9. E. SOISALON–SOININEN AND E. UKKONEN, A method for transforming grammars into $LL(k)$ form, *Acta Inform.* **12** (1979), 339–369.